WebLogic
Tengah Application Server

Not very long ago, some experts were saying that Java belonged on the client, because platform neutrality was unimportant in the server world. This turns out not to be the case. Of the 275 medium and large IT organizations recently polled by Zona Research, an overwhelming 97% plan to use Java on the server within two years.

WebLogic believes that while Java has mainly been used for standalone applets or servlets, future Java applications will be built around a distributed component architecture. Sun — with a lot of practical assistance from partners such as IBM and WebLogic — has published a comprehensive suite of Java Enterprise APIs that far surpass anything available with other languages. All that remains is for specialist Java vendors to roll out products that make full use of those APIs, and that is exactly what Tengah does.

The best software in the world is of limited value, WebLogic argues, if a proprietary design prevents it from evolving to exploit advances in technology. Real investment protection comes from open enterprise standards —the Java Enterprise APIs in Tengah's case. WebLogic's policy is to adopt industry standards as soon as they are published, and it has a good track record. dbKona was rewritten to use JDBC when it appeared, and Tengah's event management will soon be brought into line with the new Java Messaging Service. The company is agnostic about standards, supporting (for example) RMI, CORBA, and COM. Customers can choose whichever suits them best — or use them all if appropriate.

WebLogic claims that it can reduce Java deployment effort by as much as 75%, compared with in-house development, by supplying a complete

Company Profile

WebLogic, which claims to be "the oldest independent provider of exclusively Java products," was founded as recently as December 1994. This privately held San Francisco company is committed to helping developers take practical advantage of the Java standard in all its aspects. This is its mission statement:

"WebLogic brings the dynamic power of Java and the safety of the Java Enterprise Standards to the development, integration, deployment, and management of large-scale network applications."

The name of WebLogic's flagship product — Tengah — exemplifies this mission. It is the Indonesian word for "middle" or "center," and Jawa Tengah is the central administrative region of the island of Java. The whole company is fanatically committed to Java, and its expertise has led to very high productivity — which in turn, enables it to be extremely responsive to customer requests. In accordance with the best software engineering practice, the Tengah products are tested continuously (using 3,200 test cases on 20 different platforms), thus ensuring that bugs are discovered and rectified immediately.

WebLogic is growing rapidly, and now has more than 90 employees. Like its customers, most of them — even the sales and marketing staff — have a strong technical bent. With more than 1,000 customers, nobody could call it a small player — though only a minority has bought the top-end Tengah Application Server. WebLogic customers include 3M, AT&T, Baan, Bank of America, Barnes & Noble, BBC TV, Charles Schwab, Chase Manhattan Bank, Cisco, Deutsche Bank, Ericsson, the FBI, Federal Express, Fidelity Investments, Ford, Fujitsu, General Electric, IBM, JPL, John Deere, JP Morgan, Lockheed Martin, Lucent, Motorola, Nippon Steel, PeopleSoft, Reuters, Shell, SmithKline Beecham, Sony, Swiss Bank, Swiss Telecom, the US Army, US West, Walt Disney, Warner Brothers, Xerox, and Yahoo!

Tengah Application Server is bundled with Novell's Open Solutions Architecture for NetWare 5.0, and will thus be the default Java environment for Novell's customer base. Indeed, WebLogic says that about 50% of all Java ISVs currently use its software. The company's partners include Intel, Novell, Sun, Microsoft, HP, Oracle, Informix, and Cambridge Technology Partners. HP is partnering with WebLogic to market Tengah, and Intel is helping to optimize Tengah for its Pentium and forthcoming 64-bit Merced microprocessors.

WebLogic has an experienced board of directors, including Regis McKenna, Frank Caufield of Kleiner Perkins Caufield & Byers, Ali Kutay, Gary Anderson, the managing director of TL Ventures, William Seibel of Cambridge Technology Partners, and Neal Dempsey of Bay Partners. In March 1998, WebLogic secured a $12.5 million round of financing from investors including Intel, TL Ventures, Bay Partners, and Cambridge Technology Capital. This was the first time it had sought venture capital.

WebLogic has just announced that Ali Kutay, angel investor and board member since 1996, is its new president and CEO. Known as an entrepreneur, Kutay was previously founder, president, and CEO of Formtek, a Carnegie Mellon University spinoff launched in 1983 and later acquired by Lockheed.

WebLogic Inc.
550 California Street
San Francisco, CA 94104
Tel.: (415) 659-2600, (800) WebLogic
Fax: (415) 394-8619
E-mail: info@weblogic.com
WWW: http://www.weblogic.com/

Reprinted with permission

server and communication infrastructure "off the shelf." Moreover, this infrastructure is reliable, fully supported, extensible, and based on standards. And it is portable across all the leading platforms without recompilation.

Products

WebLogic's product line starts with its JDBC drivers, and culminates in the Tengah Application Server. Tengah/JDBC and jdbcKona are subsets of the full Application Server. Tengah is packaged as "embeddable infrastructure," and comes in the form of a set of Java classes. Thus, any required subset of it can be bundled with third-party software products.

The current version of the Tengah Application Server is 3.0, but 3.1 is in beta and is due to ship soon.

Tengah Application Server 3.1

WebLogic positions Tengah as "an open, extensible, standards-based platform for assembling, deploying, and managing distributed Java applications." It enables Java applications to interconnect with heterogeneous databases and other network resources, as well as other Java business components. Tengah confers security, scalable performance, and transactional integrity on the application objects under its control. It can reduce software costs and maintenance effort, because database drivers and the like only have to be installed once (on the server), rather than on every client desktop.

Tengah applications share access to services such as database connectivity, event handling, transaction control (JTS), remote method invocation (RMI), name and directory services (JNDI), security, global logging, instrumentation, and resource management.

Development Services

Although Tengah incorporates extensive development services, it does not offer its own integrated development environment (IDE). Instead, it has been designed to work in conjunction with Java IDEs, such as Symantec Visual Café, Inprise JBuilder, Sybase PowerJ, IBM VisualAge for Java, Supercede, and Microsoft Visual J++. Using such IDEs, developers can drag and drop JavaBeans to build Tengah applications.

Tengah Beans

Based on JavaBeans, Tengah Beans provides the additional ability to split a single Bean into client and server parts, giving architects the widest choice between fat, thin, or "anorexic" (HTML) clients. The JavaBeans delivered with Tengah support features, such as login, publish/subscribe, and server workspaces.

A Tengah Bean is a symmetric distributed object — its client stub and server skeleton are identical. A Tengah Bean client has no notion of server location or address. Given a server-side JavaBean, a Tengah Bean with equivalent functionality — but differently partitioned — can be instantiated. Like JavaBeans, Tengah Beans can fire events and service method invocations; however, Tengah Beans can operate as distributed objects across multiple JVMs.

Enterprise JavaBeans (EJB)

Tengah 3.0.1 has "preliminary" support for Sun's EJB (roughly speaking, the server-side version of JavaBeans). The idea is to let developers familiarize themselves with EJB. Tengah 3.1 includes a production implementation — the first, according to WebLogic — of EJB 1.0. This will enable the encapsulation of server-side business logic as secure, transactional components.

Tengah EJB supports all features of the EJB specification, including all the optional extensions:

- Entity beans with both bean-managed and container-managed persistence
- Container-managed ("automatic") persistence on files and RDBMSs
- Distributed transactions

Database Connectivity (JDBC)

Enables Java applications to access and update multiple remote databases. This functionality is described in the sections on jdbcKona and Tengah/JDBC.

Tengah Remote

Tengah Remote uses RMI to let any Java application invoke methods on objects resident in remote JVMs. It is also integrated with JNDI. This means that any client, server, or applet can register an object with the JNDI service, and any other client, server, or applet can then invoke the registered object.

Tengah RMI supports multiple transports, including improved HTTP, HTTPS, IIOP tunneling, and SSL protocols. It interoperates through IIOP and IORs, with ORBs that support the CORBA 2 standard.

Tengah COM

This facility, new in Tengah 3.1, enables any Microsoft COM component to be plugged into the Tengah framework, automatically wrapped with a Java class, and transparently shared over the network. On a practical level, this lets Solaris users run Excel and other Windows applications from their workstations.

Tengah NSAPI Plug-In

Tengah 3.1 includes a plug-in shared library to support Netscape HTTP server proxying. This means that a Netscape server can be configured to forward all servlet requests, or tunneled remote object invocations, to Tengah.

Event Management

This facility offers a true “server-push” event model for delivering information about changing conditions to interested applications, and provides publish/subscribe capabilities with server-side filtering and actions to reduce network and subscriber load. In Tengah 3.1, events are delivered in the order in which they occurred — which was not necessarily the case in previous releases.

When an event is received by the Tengah Server, it filters the event down through its “topic tree,” checking it against all registrations to find matches. If it finds a match, Tengah passes the event to the evaluator for that registration, which tests the event to determine whether it should be acted.” (Evaluators and Actions are user-written Java methods that are executed inside the Tengah Server).

Tengah events are so lightweight that a typical server can handle thousands of them every second.

Name/Directory Services (JNDI)

Through the Java Naming and Directory Interface (JNDI), Tengah can access directory services such as Novell NDS, Sun NIS+, Microsoft Active Directory, and any others supporting Lightweight Directory Access Protocol (LDAP). Tengah JNDI is a multi-tier implementation that can run within the constraints of the Java sandbox. The Tengah Server includes a naming service provider.

Servlets

Tengah implements servlets (the server equivalent of applets). Its HTTP servlets support dispatching requests to Java business logic via HTTP and dynamically constructing HTML responses. Tengah’s built-in HTTP server handles HTTP servlets as well as HTML pages, graphics, and other MIME types.

Tengah developers can build dynamic HTML pages by creating the HTML “templates” with any authoring tool, and then writing complementary Java applets. Tengah’s PageCompile servlet converts the mixed HTML and Java source files into Java servlets, which it then compiles and executes. Thereafter the conversion and compilation steps are repeated only when the sources change.

Deployment Facilities

Client to Server Communications

Tengah clients and servers communicate with each other through packet-based queues, over a transport mechanism known as RichSockets. A single connection can host Tengah/JDBC traffic, Tengah RMI traffic, and Tengah Event traffic at the same time. Tengah clients can connect over TCP, HTTP, SSL, or CORBA IIOP, with support for tunneling through firewalls. Both RMI and CORBA are supported, and COM joins them in Tengah 3.1.

Each client has its own workspace and associated send and receive queues within the Tengah server.

This arrangement confers many of the benefits of message-oriented middleware (MOM). For instance, a client may go away and reconnect — at which time it will have access to the same client workspace, and receive whatever is waiting on its receive queue.

Clocks may be synchronized between clients and servers through Tengah Time — a valuable facility in applications where exact timing is critical.

Secure Execution

Tengah provides authentication through X.509 digital certificates, communication integrity and privacy through SSL and HHTPS, and authorization through access control lists (ACLs).

Like most application servers, Tengah offers significant benefits in terms of security and its administration. To take one example, it can obviate the need to supply each client with an unencrypted user name and password for database access. A secure application server can shield the client from direct access to username and password information by providing pools of connections to each database.

Transactional Integrity

As well as DBMS transactions, Tengah supports the combination of multiple Enterprise JavaBeans into a single, unified transaction. The EJB framework generates custom classes that make an EJBean remotely accessible, intercepts all calls to the EJBean, and provides transactions, security, and persistence.

Server Clustering

WebLogic describes Tengah clustering as "a series of features to be introduced over multiple releases."

Tengah 3.1 supports request forwarding across servers, as well as proxying for dispatching into server farms. This permits load balancing at the request level. Session management and state is replicated across servers, so end-users do not usually see delays or service outages, even if a server fails. Security and systems management are also unified across each cluster. Additional clustering of EJB and RMI objects is also targeted for late 3Q98.

Performance and Scaling Capabilities

One of the first questions people ask about any Java product is "How fast does it run?" It is undeniably true that the interpretive nature of the JVM imposes a performance penalty relative to compiled native code. However, efficient design can compensate for any shortcomings in execution speed — especially in servers, which can spend much of their time waiting for network and disk I/O. WebLogic's profiling exercises have revealed that a typical application spends 80% of elapsed time "in the DBMS."

In any case, improvements in JVM design are bringing Java performance steadily closer to that of C, C++, and COBOL.

Tengah benefits from the following performance optimizations:

- *All client service requests multiplexed across a single connection.* Tengah optimizes network traffic with its clients by using an efficient packet-based, queue-based protocol.

- *Native DBMS drivers instead of ODBC.* Although ODBC provides common access to most popular RDBMSs, it does so by adopting a "lowest common denominator" approach. Tengah's native drivers take full advantage of each DBMS' special features.

- *DBMS connection pooling.* Cached connections and named cached data sets substantially reduce the number of database logins required. Connection pools enable users to share a configurable number of database connections that are established at Tengah startup.

- *DBMS query caching*. Because queries are cached on the server, the results of popular queries can be returned, without even accessing the database. Caching also helps to smooth the flow of data to clients. Tengah also supports "Named Queries," which can be invoked explicitly or triggered by any specified operation.

- *Persistent client/server connections*. Each client has its own persistent workspace within the Tengah server. A client can log back in to its previous state — including a set of cached connections to the DBMS — to continue work. Within the client workspace, objects (such as DataSets) can be saved for later use, reducing the DBMS workload and network traffic.
- *Load balancing* is implemented through dynamic relocation of application components.

In a recent benchmark, WebLogic succeeded in getting a single-processor Pentium II PC running Windows NT to handle 2,048 simultaneous clients. The server's response time was linear with the workload, and it delivered a peak throughput of nearly 800 messages per second.

Graphical Console

Tengah 3 provides a graphical systems management Console that can be run on any Java platform. Built with the JFC, the Console can be run either as an application or as an applet. It offers an impressive selection of textual and graphical views of the Tengah server, ranging from details of the class path and Java heap usage, to JDBC driver information, servlets, threads, users, workspaces, security, software versions, and licenses. In a nice touch, each display viewed is added as a tab at the top of the window so that it can easily be revisited.

In Tengah 3.1, the Console can also issue server commands, such as Start, Shutdown, Lock, and Unlock.

The Console takes about a minute to run up and authenticate the operator on a Pentium PC, but thereafter it provides acceptable response time. As with all Java applications, performance can be expected to improve substantially, and any slight inconvenience is balanced by the ability to run the Console on any client.

Integrated Logging

Tengah automatically logs diagnostic and security audit information, and provides interfaces for applications to log their own exception conditions. Optionally, HTTP access can be logged in NCSA Common Log Format.

Logs can be viewed remotely from a Web browser or from the Tengah Console.

Zero Administration Client (ZAC)

New in Tengah 3.1, the ZAC supports automatic distribution of applets and Java applications. Program libraries (even a whole new Tengah release) can be installed centrally by an administrator, and Tengah then ensures that each updated component is pushed to all appropriate machines.

Naturally, ZAC complies with the relevant W3C proposed standards such as Distribution and Replication Protocol (DRP), and Open Software Distribution (OSD).

jdbcKona

WebLogic's entry-level product, jdbcKona is a set of Type 2 JDBC drivers for Oracle, Sybase, and Microsoft SQL Server. It is shipped as part of Tengah/JDBC and Tengah Application Server.

The jdbcKona drivers are appropriate in two-tier client/server configurations, where a client computer talks directly to one or more database servers across a network. They connect to the RDBMS using vendor-supplied client libraries, which that take full advantage of the native features of each RDBMS and provide better performance than ODBC. Multiple data sources can be accessed concurrently.

Because a thin layer of C++ code is required to communicate with the client libraries, the jdbcKona drivers are the only part of Tengah that is not pure Java.

WebLogic has acquired ownership of Connect Software's Type 4 FastForward JDBC drivers for Informix and Microsoft SQL Server. These drivers have been added to the jdbcKona product line as jdbcKona/Informix4 and jdbcKona/MSSQLServer4, and are currently in beta.

dbKona

dbKona shields developers from the low-level details of JDBC by providing a higher level of abstraction. It deals in DataSets, Records, and Values, hiding the details of specific SQL statements and DBMS features. If necessary, dbKona also lets developers issue raw SQL commands with vendor-specific syntax.

A Value object encapsulates data together with its structure and type; a Java application that uses data in a Value object does not need to know anything about its structure and type. A dbKona DataSet enables a great deal of freedom in navigating through its constituent records, unlike JDBC's ResultSet, which is limited to iterating forward through records.

WebLogic provides one vendor-specific extension for dbKona: a Sequences object, for use with Oracle sequences. Other extensions may be provided in the future.

htmlKona

htmlKona provides a Java object library for generating dynamic or static HTML pages. It can be used in conjunction with jdbcKona and dbKona to generate tables from queries and integrate data (such as multimedia objects) into server-side Java classes. htmlKona supports HTML 3.2, 3.5, and 4.0, and is compatible with any Web server that supports the Java Servlet API.

Tengah/JDBC

Tengah/JDBC is a subset of the Tengah Application Server. As well as remote database access, it also offers security (SSL and ACLs), built-in HTTP servlet support, and client access using protocols such as HTTP tunneling and IIOP.

Whereas the jdbcKona drivers are designed for use in a two-tier configuration — and so include vendor-supplied client libraries — Tengah/JDBC sits on the client machine in a three-tier configuration. Because it is written in Java and uses no native methods, applets can use it to access remote databases. Another benefit of Tengah/JDBC's pure Java implementation (at least when used with jdbcKona on the server) is that clients can take full advantage of Java's multithreading facilities. This might not be the case with single-threaded third-party client libraries in the loop.

Tengah/JDBC connects to a database through a JDBC driver, which may be one of WebLogic's own jdbcKona drivers, the JDBC-ODBC gateway, or a third-party driver. The product includes the Tengah Utility Set, comprising jdbcKona, dbKona, and htmlKona. It can be used in conjunction with Tengah Secure, Tengah Console, CORBA, HTTP servlets, and HTTP firewall tunneling.

Platforms

Tengah Application Server

- *Clients* — Any platform with support for JDK 1.1 or 1.0.2
- *Servers* — Windows NT, NetWare, Unix (AIX, Digital Unix, HP-UX, SGI Irix, Solaris)

The above platforms have been fully tested. Tengah is up and running on other platforms, including Linux, other brands of Unix, Tandem, OS/400, and even OS/390. The fact that Tengah is embedded in Novell NetWare 5.0 gives a fairly broad hint that it works there too.

Pricing

Tengah Application Server

- *Development* — From $1,995 per seat
- *Deployment* — $9,995 per processor

Tengah runtime licenses cost the same flat $10K for each processor, from a Pentium-based PC server to a top-end RISC machine.

Opinion

Strengths

- Nested range of Java products from Type 2, 3, and 4 JDBC drivers, to an enterprise Java application server.
- Highly optimized for fast, efficient database access, using native drivers, pooled client and database connections, caching, and server workspaces.
- All WebLogic's products are implemented wholly in Java (apart from a few lines of interface code in jdbcKona) and are, therefore, highly portable without recompilation.
- WebLogic builds on standards wherever possible, even retrofitting its products to comply

with standards that are published after it has shipped a proprietary implementation.

- Tengah is a comprehensively equipped Java application server with support for dynamic partitioning, JDBC, JavaBeans, EJB, servlets, Java Events, JNDI, RMI, COM, CORBA, and HTTP.

- Tengah is also an HTTP server with unrivalled support for the construction of dynamic HTML pages through the use of Java and JavaScript.

- Efficient, lightweight, and secure publish/subscribe messaging service.

- Any full-featured Java IDE that supports JavaBeans can be used to develop Tengah applications.

- Tengah includes a graphical systems management console and Zero Administration Client facility that supports automatic software distribution and update.

- Tengah is used by some 50% of Java ISVs, and has been chosen by Novell to ship with every copy of NetWare 5.0.

Weaknesses

- Precisely because of its great power and flexibility, Tengah is no tool for beginners. It is best suited for experienced professional developers, with a good knowledge of Java and multi-tier distributed systems. However, WebLogic is working to improve ease of use, and features such as Tengah Beans and the EJB deployment wizard do simplify matters.

- As with other Java products currently, runtime performance may still leave something to be desired, especially on the client (i.e., if the client uses Java).

- All WebLogic's products are Java-only, which could be awkward for an organization that does all its programming in C, C++, or COBOL.

- Tengah competes with Sun's own JavaServer — an unfortunate situation that stems from Sun's decision to enter the market against its own Java partners — as well as with Greenbrier & Russel, NetDynamics, Netscape Application Server (previously Kiva), Novera, and others.

- WebLogic's future depends on the market's acceptance of the Java APIs. Many developers and ISVs are enthusiastic, but selling to mainstream IT users could be more challenging. Although some pontoons have arrived, the chasm remains to be crossed.

Conclusions

Tengah is Microsoft's nightmare come true. It cuts through all the opinions about whether Java is an operating system, a platform, a set of APIs, or just a language. Developers get a set of classes for quickly and reliably assembling three-tier applications that run unchanged across a range of leading platforms. The salient facts about Tengah are that it is pure Java, it performs virtually all of the tasks an industrial-strength application server should, and above all, it works. By that we mean that it not only performs as specified, but also that it is robust and that its runtime performance is adequate.

The fact that Tengah is written in Java is important. Fashion apart, there are very good business reasons for adopting a platform-neutral server. For instance, Tengah running on an Apple Macintosh can make a Macintosh-only DBMS accessible to JDBC clients running Windows, Unix, OpenVMS, or OS/2, as well as MacOS. And, contrary to what some pundits are saying, the promise of "write once, run anywhere" is very real. WebLogic has brought up the Tengah server — a complex piece of systems software with more than 1,000 classes — without modification, on a proprietary operating system and Java platform that had never run Tengah before, and successfully run the comprehensive test suite. The whole exercise took a couple of hours. Now *that* is portability.

Tengah Beans enable application partitioning to be delayed until deployment. This feature is very similar to the "dynamic application partitioning" introduced by Forte about five years ago, but with a couple of significant improvements. First, Tengah Beans are JavaBeans and therefore less proprietary than Forte's technique. (Of course, when Forte invented the idea there were no standards to comply with). Secondly, Tengah does not require a system generation phase.

WebLogic is well placed to fight for the lead in the emerging market for Java application servers, but it is by no means the only contender. With Java taking over as the main programming language for new projects, and three-tier architectures becoming routine for enterprise systems, it is plain that Java application servers — which combine the two — will be in huge demand.

What is surprising is how many suppliers have already sprung up. GemStone, the born-again Smalltalk ODBMS specialist, may have been first to market with its GemStone/J. But a cursory review of the field reveals at least a dozen others, including Greenbrier & Russel, Infoscape, Kiva (now Netscape Application Server), KonaSoft, NetDynamics, Netmosphere, and Novera. Then there are the heavyweights — Sun has launched its own Java Application Server, and IBM is advancing the claims of the AS/400.

In the face of all this competition, WebLogic stands well. It is the first company to offer a complete implementation of Sun's Java Enterprise APIs in the shape of an application server that is robust, scalable, and fast enough for production use. Others lag behind either in their implementation of standards, or in their ability to deliver industrial strength software. In WebLogic's own words, "Java makes it possible. Tengah makes it happen."